REMARKS

Reconsideration and allowance of this application are respectfully requested.

STATUS OF APPLICATION

Submitted herewith is a *Rescission Of Previous Nonpublication Request And Notice Of Foreign Filing*. In addition, since that Notice is being provided more than forty-five (45) days after the date of such foreign or international filing, a *Petition under 37 CFR § 1.137(b)* is being filed herewith.

Claims 1-65 are pending in this application.

By this Amendment, the specification has been amended to correct a typographical error in paragraph 0032 (to insert a missing space between the words "information" and "regarding").

No new matter has been added by this amendment.

PRIOR ART REJECTIONS

This invention relates to managed object replication and delivery. As noted in the application, a

> typical content delivery network (CDN) operator deploys one or more parent servers, hosting a plurality of objects, in a network and one or more edge servers at the edge of the network to facilitate more cost-effective and efficient delivery of such objects to an end-user (client). End-users or client proxies that access customers' objects are called clients. Content provider companies, organizations, etc. that subscribe to the CDN service are referred to as customers. * * * It is typically desirable to serve objects from edge servers because the edge servers are typically closer (by various measures of distance) to end-users. *Specification,* ¶0010.

Understanding the problems of edge servers, the inventors were the first to realize the value of populating CDN caches with *popular* content. See, e.g., the application:

> It is typically not feasible to store all objects on the edge servers. The main difficulty is due to the fact that many such objects are very large (typically on the order of 10 MB (10,000,000 bytes) -- in the neighborhood of 500 MB for movies). The storage and rack space required to accommodate often large and sometimes rarely requested objects at every edge server can be cost prohibitive as the number of customers grows and the number of their objects increases. *Specification* ¶0011.

Accordingly, in some aspects, this invention provides for methods and systems that intelligently replicate objects to edge servers if the objects are popular enough. Likewise, an object may be is removed from an edge server when the object is no longer popular.

The Examiner has rejected claims 1-65 under 35 U.S.C. § 102(e) as being anticipated by Jungck. The grounds for this rejection are respectfully traversed.

Claim 1 and its dependents recite a method for managed object replication and delivery. The method comprises directing a request by a client for an object to an edge server in a network. If the edge server has the requested object, serving the requested object to the client; otherwise, redirecting the client request to a server that has the requested object and serving the requested object to the client. The method further recites "*if the requested object is popular*, replicating the requested object to the edge server."

Jungck relates to the "virtual edge placement of web sites." Jungck does describe cache servers and caching, but there is nothing in Jungck to teach or in any way suggest the claimed method of replicating a requested object "*if the requested object is popular*." The cache servers described by Jungck (at paragraphs 57-59) are nothing more than regular cache servers that cache all

objects. As noted in Jungck, "Where the requested content is not in the cache (also known as a "miss"), the cache forwards the request onto the content source. When the source responds to the request by sending the content to the client 102, 104, 106, the cache server 208 saves a copy of the content in its cache for later requests." *Jungck* ¶0057. Thus, in Jungck, objects are cached based merely on the fact that they have been requested – based on the current request.

Jungck actually teaches away from caching based on popularity. The only caching "policies" described in Jungck are based on temporal or spatial locality. As described in Jungck, "[c]aches operate on two principles, temporal locality and spatial locality. Temporal locality is a theory of cache operation which holds that data recently requested will most likely be requested again. This [temporal locality] theory dictates that a cache should store only the most recent data that has been requested and older data can be eliminated from the cache. Spatial Locality is a theory of cache operation which holds that data located near requested data (e.g. logically or sequentially) will be likely to be requested next. This theory dictates that a cache should fetch and store data in and around the requested data in addition to the requested data." *Jungck ¶0058.*

Based on Jungck's teachings, if his caches operate based on temporal locality, "a cache should store only the most recent data that has been requested and older data can be eliminated from the cache." On the other hand, if his caches operate on spatial locality, "a cache should fetch and store data in and around the requested data in addition to the requested data." (For spatial locality Jungck is silent about cache replacement policies.) But, regardless of whether Jungck's caches use spatial or temporal replacement, he is silent about any popularity-based caching.

Using Jungck, an unpopular object might be cached merely because it has been requested, and with no regard to its popularity. Suppose, e.g., that there are two objects, O_1 and O_2, with O_1 being very popular and O_2 never having been

requested. According to the present invention recited in claim 1, if O_1 was requested and not available at an edge server, then, because of its popularity, O_1 would be replicated at the edge server. The object O_2, on the other hand, would not be replicated based on its popularity (it may be replicated for some other reason). This is particularly important in the context of streamed content such as movies. As noted in the application, a movie may be on the order of 500 Mbytes. ¶0011. It is desirable to have popular movies replicated at the edge servers, but the same does not apply to non-popular movies. Under Jungck's scheme, on the other hand, a single request for a movie – perhaps the first request ever – will cause that movie to be cached, possibly knocking a much more popular movie out of the cache.

While the replication scheme of present invention is well suited to streaming media, Jungck admits that caching (as he would implement it) is not so suited. "[C]ache servers 208 often cannot support the bandwidth and processing requirements of streaming media, such as video or audio, and must defer these content requests to the server 108 which are the source of the content." *Jungck ¶0059.*

Since Jungck fails to teach or suggest at least one claimed element, claim 1 and its dependents are patentable over Jungck. Similar arguments apply to the other independent claims 16, 23, 38, 45 and 59 and their dependents, which are therefore also all patentable over Jungk.

There are other differences between the teachings of Jungck and the claims, some of which are further discussed here:

Further as to claims **3, 25 and 47**, applicant respectfully submits that Jungck neither teaches nor in any way suggests the claimed "recursively redirecting the request until a parent server in the network having the requested object is reached and serving the requested object to the client from the parent

server." Jungck is completely silent about any sort of recursive redirection until a server having a requested object is found.

Still further as to claims **6, 28 and 50**, applicant respectfully submits that Jungck neither teaches nor in any way suggests a method or system in which a best or optimal server is selected based on factors such as "the likelihood of a copy of the requested object being available at the edge server, and the bandwidth between the edge server and the client." In this case the Examiner did not cite to any particular part of Jungck.

Still further as to claims **7, 29 and 51**, applicant respectfully submits that Jungck neither teaches nor in any way suggests the claimed "replicating the requested object to the edge server comprises replicating the requested object to the edge server from a parent server." Jungck has no notion of parent servers, and so cannot teach this replicating.

Still further as to claims **8, 9, 30, 31, 52, 53 and 65**, applicant respectfully submits that Jungck neither teaches nor in any way suggests the claimed popularity-based replication. Jungck has no notion of popularity or parent servers.

Still further as to claims **10, 21, 32, 43, 54 and 63**, applicant respectfully submits that Jungck neither teaches nor in any way suggests the claimed "wherein whether the requested object is popular is determined using at least a request rate for the requested object." The Examiner again cites Jungck ¶0058, but again there is nothing in that paragraph or anywhere else in Jungck to teach or in any way suggest any measure of popularity, let alone popularity being determined as claimed, "using at least a request rate for the requested object." In fact, here again, Jungck teaches away from the claimed invention. In Jungck, all requested objects are cached (and if spatial-locality-based caching is used, then objects "near" requested objects are also cached (see Jungck ¶0058). But Jungck does not teach or suggest making any use whatsoever of the request rate of an object.

Still further as to claims **11, 12, 17, 18, 33, 34, 39, 40, 55 and 60**, applicant respectfully submits that Jungck neither teaches nor in any way suggests deleting objects that are no longer popular. As noted above, Jungck has no notion of popularity. According to Jungck, he will delete the oldest objects from the cache, regardless of their popularity ("theory dictates that a cache should store only the most recent data that has been requested and older data can be eliminated from the cache." ¶0058). Jungck may well delete the most popular object from the cache. Jungck may delete an unpopular item from a cache, but not because of its popularity – merely because it is the oldest item in the cache.

Still further as to claims **13, 19, 35, 41, 56 and 61**, applicant respectfully submits that Jungck neither teaches nor in any way suggests the claimed "replicating the requested object in accordance with a dynamic replication threshold." (Neither the word "threshold" nor any such concept appears in Jungck.)

Still further as to claims **14, 20, 36, 42, 57 and 62**, applicant respectfully submits that Jungck neither teaches nor in any way suggests the claimed deletion of least popular objects until there is enough storage for a popular object. As recited in, e.g., claim 14, the invention comprises "replicating the requested object when a popularity of the requested object is greater than a threshold popularity and there is enough storage to replicate the requested object." Jungck is silent about popularity in general and definitely about popularity of a requested object exceeding a threshold popularity. Then, as to the case when there is not enough storage to replicate a requested object, there is absolutely nothing in Jungck about repeatedly deleting the least popular object from the storage, until enough storage is available for the requested object or the popularity of the requested object is less than the popularity of the least popular object in the storage.

Here, again, the Examiner cites Jungck ¶57, but, as with the remainder of Jungck, that section is silent as to any measure of popularity, let alone any cache deletion / filling technique based on popularity.

Still further as to claims **15, 22, 37, 44, 58 and 64**, applicant respectfully submits that Jungck neither teaches nor in any way suggests the claimed serving "wherein serving the requested object is performed separately from replicating the requested object." The section of Jungck cited by the Examiner, ¶¶25 and 27, describe an architecture of a network (¶25) and the manner in which content is served (¶27), but there is nothing in Jungck about replication, let alone about the timing of such replication.

In view of the above, withdrawal of this rejection under § 102 is respectfully requested.

CONCLUSION

Applicant respectfully submits that all claims are in condition for allowance and an early action to that effect is earnestly solicited. The Examiner is invited to contact the undersigned at the number provided to resolve any outstanding issues.

Deposit Account No.: 501860 Order No. (Client-Matter No.): 2615-0040

CHARGE STATEMENT: The Commissioner is hereby authorized to charge any fee specifically authorized hereafter, or any missing or insufficient fee(s) filed, or asserted to be filed, or which should have been filed herewith or concerning any paper filed hereafter, and which may be required under Rules 16-18 (missing or insufficiencies only) now or hereafter relative to this application and the resulting Official document under Rule 20, or credit any overpayment, to our Account/Order Nos. shown above, for which purpose a duplicate copy of this paper is attached.

This Charge Statement does not authorize charge of the issue fee until/unless an issue fee transmittal form is filed.

CUSTOMER NUMBER
42624

Respectfully submitted,

By: _____
Brian Siritzky
Registration No.: 37,497

DAVIDSON BERQUIST JACKSON & GOWDEY, LLP
703.894.6400